Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 37 (2014) 109 – 116

Procedia Computer Science

The 5th International Conference on Emerging Ubiquitous Systems and Pervasive Networks (EUSPN-2014)

Frequent Pattern Mining Algorithms for Finding Associated Frequent Patterns for Data Streams: A Survey

Shamila Nasreen[a], Muhammad Awais Azam[b], Khurram Shehzad[a], Usman Naeem[c], Mustansar Ali Ghazanfar[a]

[a] *Department of Software Engineering, UET Taxila, 47080, Pakistan*
[b] *Department of Computer Engineering, UET Taxila, 47080, Pakistan*
[c] *School of Architecture, Computing and Engineering, University of East London, United Kingdom*

Abstract

Pattern recognition is seen as a major challenge within the field of data mining and knowledge discovery. For the work in this paper, we have analyzed a range of widely used algorithms for finding frequent patterns with the purpose of discovering how these algorithms can be used to obtain frequent patterns over large transactional databases. This has been presented in the form of a comparative study of the following algorithms: Apriori algorithm, Frequent Pattern (FP) Growth algorithm, Rapid Association Rule Mining (RARM), ECLAT algorithm and Associated Sensor Pattern Mining of Data Stream (ASPMS) frequent pattern mining algorithms. This study also focuses on each of the algorithm's strengths and weaknesses for finding patterns among large item sets in database systems.

Peer-review under responsibility of the Program Chairs of EUSPN-2014 and ICTH 2014.

Keywords: Frequent Pattern Growth (FP Growth), Rapid Association Rule Mining (RARM), Data Mining, Frequent Patterns

1. Introduction

Frequent pattern mining has been an important subject matter in data mining from many years. A remarkable progress in this field has been made and lots of efficient algorithms have been designed to search frequent patterns in a transactional database. Agrawal et al. (1993) firstly proposed pattern mining concept in form of market based analysis for finding association between items bought in a market. This concept used transactional databases and other data repositories in order to extract association's casual structures, interesting correlations or frequent patterns among set of [1]. Frequent patterns are those items, sequences or substructures that reprise in database transactions with a user specified frequency. An itemset with frequency greater than or equal to minimum threshold will be considered as a frequent pattern. For example in market based analysis if the minimum threshold is 30% and bread appears with eggs and milk more than three times or at least three times then it will be a frequent itemset [2].

Frequent pattern mining can be used in a variety of real world applications. It can be used in super markets for selling, product placement on shelves, for promotion rules and in text searching. It can be used in wireless sensor networks especially in smart homes with sensors attached on Human Body or home usage objects and other applications that require monitoring of user environment carefully that are subject to critical conditions or hazards such as gas leak, fire and explosion [3].These frequent patterns can be used to monitor the activities for dementia patients. It can be seen as an important approach with the ability to monitor activities of daily life in smart environment for tracking functional decline among dementia patients [4].

* Shamila Nasreen. Tel.: +92-05827-961016; fax: +92-05827-961016.
E-mail address: Shamila_nasreen131@yahoo.com

Peer-review under responsibility of the Program Chairs of EUSPN-2014 and ICTH 2014.
doi:10.1016/j.procs.2014.08.019

In mining pattern stage different techniques are applied to find candidates for frequent patterns and then frequent patterns are generated. There are two main problems with frequent pattern mining techniques. First problem is that the database is scanned many times, second is complex candidate generation process with too many candidate itemset generated. These two problems are efficiency bottleneck in frequent pattern mining. Studies demonstrate that a lot of efforts have been performed for devising best techniques and worth mentioning approaches are Apriori, RARM, ECLAT, FP Growth and ASPMS algorithms.

2. Literature Review

Several algorithms for mining associations have been suggested in the literature work [5] [6] [7] [8] [3] [9] [10] [11] [12] The Apriori algorithm [5] is most widely used algorithm in the history of association rule mining that uses efficient candidate generation process, such that large Itemset generated at k level are used to generate candidates at $k+1$ level. On the other hand, it scans database multiple times as long as large frequent Itemsets are generated. Apriori *TID* generates candidate Itemset before database is scanned with the help of Apriori-gen function. Database is scanned only first time to count support, rather than scanning database it scans candidate Itemset. This variation of Apriori performs well at higher level where as the conventional Apriori performs better at lower levels [6]. Apriori Hybrid is a combination of both the Apriori and Apriori *TID*. It uses apriori *TID* in later passes of database as it outperforms at high levels and Apriori in first few passes of database. DHP (Direct hashing and Pruning) [7] tries to maximize the efficiency by reducing the no of candidates generated but it still requires multiple scans of database. DIC [8] based upon dynamic insertion of candidate items, decrease the number of database scan by dividing the database into intervals of particular sizes. CARMA(Continuous Association Rule Mining Algorithm) proposed in [9] generates more candidate Itemset will less scan of database than Apriori and DIC, however it adds the flexibility to change minimum support threshold. ECLAT [10] with vertical data format uses intersection of transaction ids list for generating candidate Itemset. Each item is stored with its list of Transaction ids instead of mentioning transaction ids with list of items. Sampling algorithm chokes the limitation of I/O overhead by scanning only random samples from the database and not considering whole database. Rapid Association Rule mining (RARM) proposed in [11] generates Large *1- Itemset* and large *2-Itemset* by using a tree Structure called *SOTrieIT* and without scanning database. It also avoids complex candidate generation process for large *1-Itemset* and Large *2-Itemset* that was the main bottleneck in Apriori Algorithm.

Another accomplishment in the development of association rule mining and frequent pattern mining is FP-Growth Algorithm which overcomes the two deficiencies of the Apriori Algorithm [1]. Efficiency of FP-Growth is based on three salient features: (1) A divide-and-conquer approach is used to extract small patterns by decomposing the mining problem into a set of smaller problems in conditional databases, which consequently reduces the search space (2) FP-Growth algorithm avoid the complex Candidate Itemset generation process for a large number of candidate Itemsets, and (3) To avoid expensive and repetitive database scan, database is compressed in a highly summarized, much smaller data structure called FP tree [12]. In [3] a novel tree structure is proposed, called associated sensor pattern stream tree (ASPS-tree) and a new technique, called associated sensor pattern mining of data stream (ASPMS), using sliding window-based associated sensor pattern mining for Wireless Sensor Networks. ASPMS algorithm can extract associated sensor patterns in the current window with frequent pattern (FP)-growth like pattern-growth method after getting useful information from the ASPS-Tree.

3. Comparative Study of Pattern Mining Techniques

Frequent pattern mining techniques have become an obvious need in many real world applications e.g. in market basket analysis, advertisement, medical field, monitoring of patients routines etc. To make a comparison among these algorithms, we use the same transactional database for all algorithms, this transactional database is based on data of a smart home where sensors are installed on daily usage objects and patients while performing their daily tasks, touch these objects and these sensor items are maintained in database. Studies of Frequent pattern mining is acknowledged in the data mining field because of its applicability in mining sequential patterns, structural patterns, mining association rules, constraint based frequent patterns mining, correlations and many other data mining tasks. Efficient algorithms for mining frequent Itemsets are crucial for mining association rules as well as for some other information mining assignments [13]The Problem of mining frequent itemset ascended first as sub-problem of mining association rules [5].

A database consists of transactions and a transaction is denoted by T. Let there is an itemset *I*= {*I1,I2…In*} consist of n items. A Transaction *T* contains a subset of items from itemset *I*. Association rule is in the form of inference stating such that if x then y(x→ y) where x and y both are subset of items in the Itemset *I*. As transactional database is large and we are interested in those items that are used frequently, there is an important parameter "support" that helps in identifying those items that are of interest. Support for an association rule (x→ y) is defined as no of transactions or percentage which contains xUy over total number of transactions in database. Minimum lower bound of support for association rule is set by user and this support value is set as a minimum threshold and itemset whose number of transactions is above than defined threshold is considered as frequent itemset. As this threshold is a large value, more valuable knowledge is obtained and if this threshold is a minimum value, a large number of Itemsets are generated. Therefore irrelevant information should be pruned, that is the main goal of frequent pattern mining. In order to analyze different frequent pattern mining algorithms in coming paragraphs comparative analysis of these algorithms have discussed with the purpose to investigate their strengths and weaknesses in order to utilize their effectiveness in respective field.

3.1 Apriori Algorithm

Agrawal and Srikant (1994) firstly proposed Apriori algorithm. This algorithm is based on Apriori property which states "every sub *(k-1)*-Itemset of frequent *k-Itemset* must be frequent" [1].Two main process are executed in apriori algorithm: one is candidate generation process, in which the support count of the corresponding sensor items is calculated by scanning transactional database and second is large itemset generation, which is generated by pruning those candidate Itemsets which has a support count less than minimum threshold. These processes are iteratively repeated until candidate Itemsets or large Itemsets becomes empty as in example shown in Fig 1. Original database is scanned first time for the candidate set, consists of one sensor item and there *support* has counted, then these 1-*Itemset* candidates are pruned by simply removing those items that has an item count less than user specified threshold (in above case threshold=30%). In second pass database is scanned again to generate 2-*Itemset* candidates consist of two items, then again pruned to produced large 2-*Itemset* using apriori property. According to apriori property every sub 1-*Itemset* of 2 frequent Itemsets must be frequent. This process ends as in fourth scan of database 4- *Itemset* candidate will be pruned and large itemset will be empty.

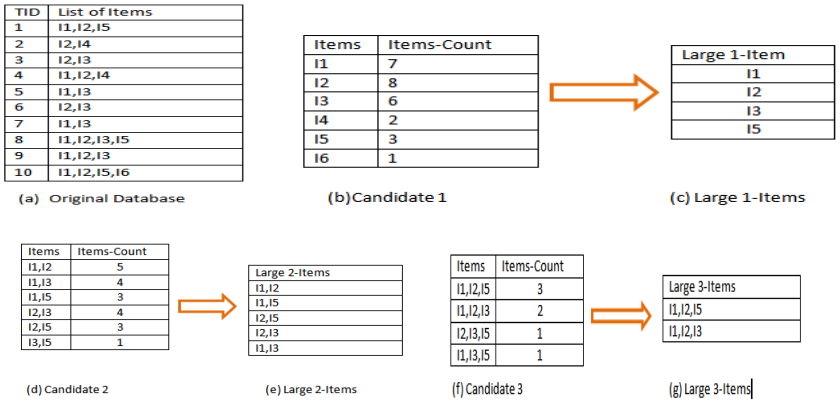
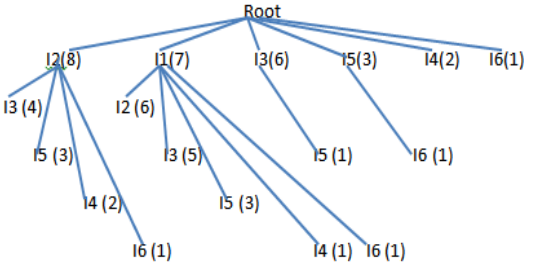


Fig 1: Apriori Process of Mining Patterns

There are two limitations of this algorithm: one is complex candidate itemset generation process which consumes large memory and enormous execution time and second problem is excessive database scans for candidate generation. Generally there are two ways to overcome these limitations: one way is to explore different pruning and filtering techniques to make candidate *Itemset* smaller. Second approach is either replace original database with subset of transaction based on large frequent *Itemset* or minimizes the number of scans over the database [14].

3.2 Rapid Association Rule Mining (RARM)

RARM is an algorithm that avoids complex candidate generation process. By using a novel tree data structure known as *Support-Ordered Trie Itemset* (SOTrieIT), RARM accelerates the mining process even at low *support* thresholds [11]. A *TrieIT* is a set of nodes in tree consisting of 2 values (Item Label and Item Support). *SOTrieIT* is a sorted ordered *TrieIT* in which nodes are sorted with their respective *support* count. Highest support item moves to the left most nodes and lowest support node is at the right most position in the tree. To Construct *SOTrieIT* tree structure, from each transaction only *1-Itemset* and *2-Itemset* are extracted. For example from TID 1={I_1,I_2,I_5} possible information extracted will be {(I_1),(I_2),(I_5),(I_1,I2),(I_1,I5),(I_2,I5)}.Those items that have already exist in the tree, their *support* count is increased by 1, whereas items that are currently do not exist in tree, create a node with label of item name and *support* count 1.After Constructing *SOTrieIT* tree, it is sorted according to their respective *support* count i.e. Fig 2 (a). Level one of *SOTrieIT* generates *1-Itemset* and level 2 generates *2-Itemset*.Depth first search can be used to traverse tree to generate *1-Itemset* and *2-Itemset*. When *1-Itemset* and *2-Itemset* are generated, large itemset can be generated from these two Itemsets using Apriori Algorithm (Fig 2 (b)) that will ultimately reduce execution time of candidate generation process.



(a) Rapid Association mining Rule (RARM)

Applying priori algorithm after generating 1-itemset & 2-itemset

C3 items	Count No	
I1,I2,I5	3	
I1,I2,I3	2	
I2,I3,I5	1	x
I1,I3,I5	1	x

Large Itemset L3
I1,I2,I5
I1,I2,I3

(b) Frequent Pattern & candidate 3-Itemset

Fig 2: Rapid Association Rule Mining Process

As the main bottleneck in association rule mining and in Apriori algorithm is the generation of candidate 1-*Itemset* and 2-*Itemset*, Performance of RARM is radically improved by using *SOTrieIT*. Generating 1-*Itemset* and 2-*Itemset* is a time consuming process in data mining and candidate 1-*Itemset* and 2-*Itemset* can easily be extracted from the *SOTrieIT* [11]. RARM also have two limitations. It is difficult to use RARM in interactive mining because if the user support threshold is changed, the whole process will have to repeat. RARM is also not suitable for incremental mining, as database size is continuously increasing with addition of new transaction; whole process needs to repeat again and again.

3.3 Equivalence CLAss Transformation (ECLAT)

ECLAT algorithm uses vertical database format whereas in Apriori and RARM horizontal data format (*TranscationId, Items*) has been used, in which transaction ids are explicitly listed. While in vertical data format (*Items, TransactionId*) Items with their list of transactions are maintained. ECLAT algorithm with set intersection property uses depth-first search algorithm [15]. All frequent *Itemsets* can be computed with intersection of *TID*-list [10]. In first scan of database a *TID* (*TranscationId)* list is maintained for each single item. *k*+1 Itemset can be generated from *k* Itemset using apriori property and depth first search computation. (*k+1)-Itemset* is generated by taking intersection of *TID*-set of frequent *k-Itemset* [2].This process continues, until no candidate *Itemset* can be found (as shown in Fig 3).

One advantage of ECLAT algorithm is that to count the support of *k+1* large *Itemset* there is no need to scan the database; it is because support count information can be obtained from *k* Itemsets. This algorithm avoids the overhead of generating all the subsets of a transaction and checking them against the candidate hash tree during support counting [16].

Itemset	TID
I1	1,4,5,7,8,9,10
I2	1,2,3,4,6,8,9,10
I3	3,5,6,7,8,9
I4	2,4
I5	1,8,10
I6	10

(a) 1-Itemset in VDF

Itemset	TID
I1,I2	1,4,8,9,10
I1,I3	5,7,8,9
I1,I4	4 ×
I1,I5	1,8,10
I2,I3	3,6,8,9
I2,I4	2,4
I2,I5	1,8,10
I3,I5	8 ×
I5,I6	10 ×

(b) 2-Itemset in VDF

Itemset	TID
I1,I2,I3	8,9
I1,I2,I4	4 ×
I1,I2,I5	1,8,10
I1,I3,I5	8 ×

(c) 3-Itemset in VDF

Fig 3: ECLAT Algorithm

3.4 Frequent Pattern (FP) Growth Algorithm

In the field of data mining, the most popular algorithm used for pattern discovery is FP Growth algorithm. To deal with the two main drawbacks of Apriori algorithm in [12] a novel, compressed data structure named as FP-tree is constructed, which is prefix-tree structure storing quantifiable information about frequent patterns. Based on FP tree a frequent pattern growth algorithm was developed.

It's a two-step approach. In first step a frequent pattern tree is constructed scanning database twice. In first pass of database, data is scanned and support count for each item is calculated, infrequent patterns are deleted from the list and remaining patterns are sorted in descending order. In 2nd pass of database, FP Tree is build. In 2nd step using FP growth algorithm frequent patterns are extracted from FP Tree.

Conditional FP tree base and Conditional FP tree are based on *node link property* and *prefix path property*. Conditional pattern base for each element in head table is shown in Fig 4(a). Conditional tree constructed for *I5* is shown in Fig 4(b). Conditional FP tree is constructed for the frequent items of pattern base. Once Conditional FP tree is constructed, frequent patterns need to be extracted as given in Table 1.

FP growth algorithm is good in achieving three important objectives; first is that of which is that database is scanned only two times and computational cost is decreased dramatically. Second main objective is that no candidates itemset are generated. Third objective is that it uses divide and conquer approach which consequently reduces the search space. On the other hands FP growth algorithm has one drawback. It is difficult to use in incremental mining, as new transactions are added to the database, FP tree needs to be updated and the whole process needs to repeat.

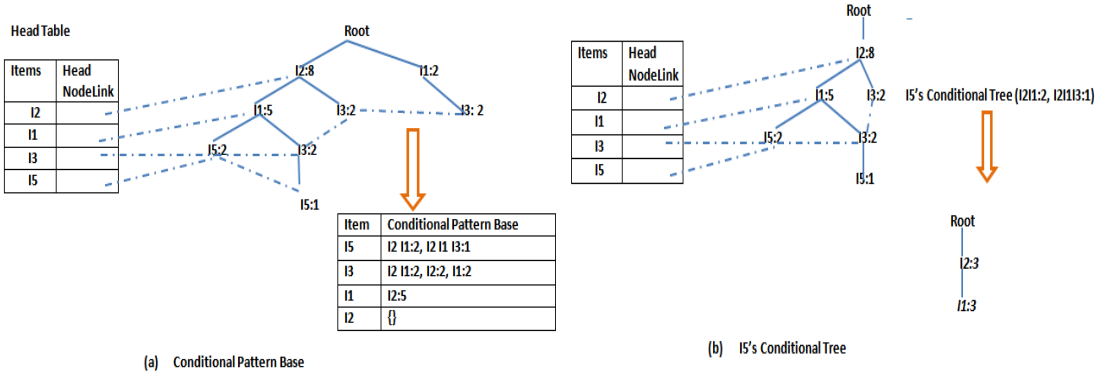


Fig 4: Conditional pattern Base & Conditional FP Tree

Table 1: Conditional FP Tree and Associated Frequent Patterns

Item	Conditional Tree	Frequent Pattern
*I*5	{(*I*2 *I*1)}| *I*5	*I*2 I5, *I*1 I5 , *I*2 *I*1 *I*5
*I*3	{(*I*2 *I*1)}| *I*3	*I*2 I3 , *I*1 I3 , *I*2 *I*3 *I*5

3.5 Associated sensor pattern mining of data stream ASPMS

ASPMS is a new technique designed to find frequent patterns among sensors in sensor wireless network. It uses an innovative tree structure called associated sensor pattern stream tree (*ASPS-tree*). In which a single scan of database used and then associated sensor pattern mining of data stream (ASPMS) algorithm is used for sliding window based associated sensor patterns for mining. This algorithm can capture important knowledge from the stream contents for the current window of the sensor in a batch-by-batch manner. Inside the nodes of an *ASPS-tree* in a sensor appearance order and then restructure the tree in a frequency-descending order. Then finally compress the tree by merging the same support sensor node in a single node in each branch of the tree [3]. This technique has two phases: Tree construction and mining associated sensor patterns. Tree construction phase is further divided into two sub phases; insertion phase and restructuring & compressing phase.

Original database is divided into equal size of windows. Each window contains equal number of batches and each batch contains equal no of transactions. Original database as shown in Fig 5 is divided into 3 equal size of windows. Windows 1 contain batch 1, batch 2 and batch 3. Batch 1 contains *TID1& TID2*. Complete tree for all transactions in windows 1 is shown in Fig 6(a). Before transactions of windows 2 are inserted batch by batch, tree needs to be compressed and restructured.

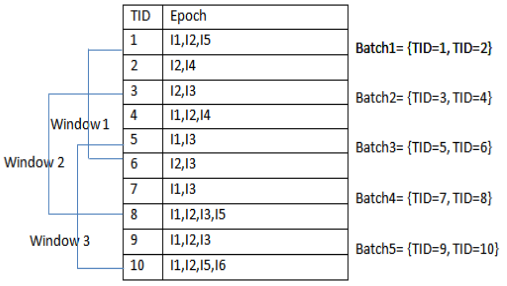


Fig 5: Original Database of sensor data streams.

Sort the SO-List in support descending order and restructure tree according to SO-list using merge sort. If two nodes of same branch have same support count, merge them into single node using Branch sort Method (BSM). Batch 1 is deleted, Sort SO-List in descending order according to its support count, restructure the tree. Complete ASPS tree after inserting windows 3 transactions batch by batch and compressing tree has shown in Fig 6(b). Next step is to extract associated frequent sensor patterns from the ASPS tree.

Conditional pattern base and conditional tree is same as in FP growth algorithm. Generated associated sensor pattern extracted from conditional tree for I5 are shown in Table 2. ASPMS requires less memory as it uses a compact tree structure which is highly compressed using branch merge sort . It requires only single scan of database. It is also highly suitable for interactive mining. As this approach is based on sliding windows, older information is deleted and new information will be considered. This approach sustain this feature that new transactions can be easily added to the tree and then tree can easily be restructured.

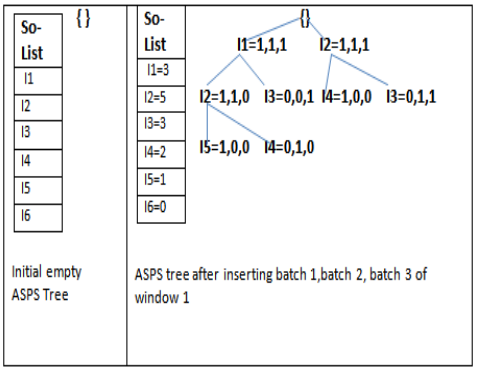


Fig 6 (a) Windows 1 (batch 1,2,3 Insertion)

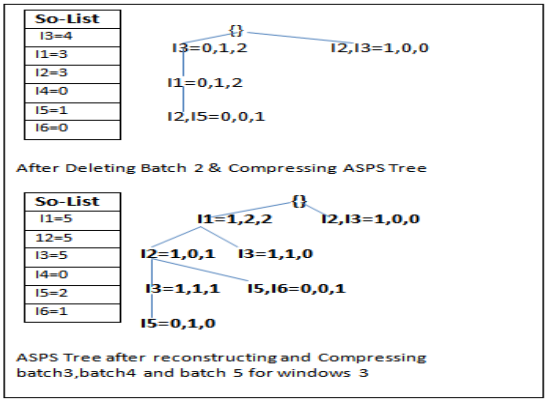


Fig 6 (b) ASPS Tree for windows 3

Table 2: Mining the ASPS Tree and associated sensors patterns

Data Items	Conditional Pattern Base	Conditional tree	Associated Sensor Patterns
*I*5	{(*I*1 *I*2 *I*3:1),(*I*1 *I*2:1)}	<*I*1 *I*2:2>	*I*1 *I*5:3 ,*I*2 *I*5:3 ,*I*1 *I*2 *I*5:3
*I*3	{(*I*1 *I*2:1),(*I*1:1),(*I*2:1)}	<*I*1 *I*2:2>	*I*1 *I*3:3 , *I*2 *I*3:4 , *I*1 *I*2 *I*3:2

4. Discussion

The comparative analysis provides a framework that clearly shows the technique, database scan, and execution time of various frequent pattern mining algorithms. The above mentioned algorithms also compared on the basis of used data format and storage structure.

The performance of any algorithm can be estimated by the number of required database scan to extract patterns. The storage consumption of different algorithms can be assessed for their utilization of memory to generate less candidate Itemset or avoid candidate Itemset generation process. Table 3 clearly illustrate that ASPMS overtake other algorithms as it perform single scan of database and more flexible for addition and deletion of transactions. ASPMS also utilize less memory as it compress the same frequency nodes into a single node using BSM.ECLAT algorithm is better than Apriori and near equivalent to FP Growth. RARM performs better at various support thresholds and is scalable. RARM performs better than Apriori and FP growth algorithm in terms of its execution time, however FP growth requires less database scan as compared to RARM algorithm.

Table 3: Comparative Analysis of Pattern Mining Algorithms

	Apriori	RARM	ECLAT	FP-Growth	ASPMS
Technique	Breadth first search & Apriori property (for Pruning)	Depth first search on SOTrieIT to generate 1-Itemset & 2-Itemset.	Depth first Search & Intersection of transaction ids to generate candidate itemset.	Divide and conquer	BSM(Branch Sort Method) using merge sort.
Database Scan	Database is scanned for each time a candidate item set is generated	Database is scanned few times to construct a SOTrieIT Tree structure.	Database is scanned few times(Best case=2)	Database is scanned two times only	Database is scanned only One time
Time	Execution time is considerable as time is consumed in scanning database for each candidate item set generation	Less execution time as compared to Apriori Algorithm and FP Growth algorithm	Execution time is less then apriori algorithm	Less time as compared to Apriori algorithm	Less execution time as compared to FP growth algorithm
Drawback	Too many Candidate Itemset. Too many passes over database. Requires large memory space.	Difficult to use in interactive system mining Difficult to use in incremental Mining	It requires the virtual memory to perform the transformation	FP-Tree is expensive to build Comsumes more memory.	
Advantage	Use large itemset property. Easy to Implement.	No candidate generation. Speeds up the process for generating candidate 1-Itemset & 2-Itemset.	No need to scan database each time a candidate Itemset is generated as support count information will be obtained from previous Itemset.	Database is scanned only two times. No candidate generation	Highly suitable for interactive mining. It requires less memory because of its compression feature in ASPs tree.
Data Format	Horizontal	Horizontal	Vertical	Horizontal	Horizontal
Storage Structure	Array	Tree	Array	Tree(FP Tree)	Tree(ASP Tree)

5. Conclusion

We have comparatively analyzed various frequent pattern mining algorithms like Apriori, RARM, ECLAT, FP Growth and ASPMS in mining association rule and data mining. We also have compared these algorithms using same database transactions to understand their edge properties. Major issues in this context were how to avoid complex candidate generation process, large number of database scans and execution time and memory requirements for large transactional database and found ASPMS perform well as compared to other algorithms.

REFERENCES

[1] Sourav S. Bhowmick Qiankun Zhao, "Association Rule Mining: A Survey," Nanyang Technological University,Singapore, 2003.

[2] Jiawei Han · Hong Cheng · Dong Xin · Xifeng Yan, "Frequent pattern mining: current status and future Directions," *Data Mining Knowl Discov*, vol. 15, no. I, p. 32, 2007.

[3] Iqbal Gondal and Joarder Kamruzzaman Md. Mamunur Rashid, "Mining Associated Sensor Pattern for data stream of wireless networks," in *PM2HW2N '13*, Spain, 2013, p. 8.

[4] M.A. Azam and Loo, J. and Naeem, Usman and Khan, S.K.A. and Lasebae, A. and Gemikonakli Azam, "A Framework to Recognise Daily Life Activities with Wireless Proximity and Object Usage Data," in *3rd IEEE International Symposium on Personal, Indoor and Mobile Radio Communication 2012.*, Sydney, Australia, 2012, p. 6.

[5] Imielienskin T. and Swami A. Agrawal R., "Mining Association Rules Between set of items in largedatabases," in *Management of Data*, 1993, p. 9.

[6] H. Mannila, R. Srikant, H. Toivonen, and A. Inkeri R. Agrawal, "Fast Discovery of Association Rules," in *Advances in Knowledge Discovery and Data Mining*, 1996, pp. 307-328.

[7] M. Chen, and P.S. Yu J.S. Park, "An Effective Hash Based Algorithm for Mining Association Rules," in *ACM SIGMOD Int'l Conf. Management of Data*, May, 1995.

[8] R. Motwani, J.D. Ullman, and S. Tsur S. Brin, "Dynamic Itemset Counting And Implication Rules For Market Basket Data," *ACM SIGMOD, International Conference on Management of Data*, vol. 26, no. 2, pp. 55–264, 1997.

[9] C. Hidber, "Online Association Rule Mining," *ACM SIGMOD International Conference on Management of Data*, vol. 28, no. 2, pp. 145–156, 1999.

[10] Mohammed J. Zaki, "Scalable Algorithms for Association Mining," *IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING,* pp. 372-390, 2002.

[11] WeeKeong,YewKwong Amitabha Das, "Rapid Association Rule Mining," in *Information and Knowledge Management*, Atlanta, Georgia, 2001, pp. 474-481.

[12] Jian Pei, Jiawei Han, "Mining Frequent patterns without candidate generation," in *SIGMOD '00 Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, New York, NY, USA, 2000, pp. 1-12.

[13] Panchal Mayur, Ladumor Dhara, Kapadiya Jahnvi, Desai Piyusha, Patel Tushar S., "An Analytical Study of Various Frequent Itemset Mining Algorithms," *Research Journal of Computer and Information Technology Sciences*, p. 4, 2013.

[14] Chistopher.T, PhD Saravanan Suba, "A Study on Milestones of Association Rule Mining ," *International Journal of Computer Applications*, p. 7, June 2012.

[15] Borgelt C., "Efficient Implementations of Apriori and Eclat," in *1st IEEE ICDM Workshop on Frequent Item Set*, 2003, p. 9.

[16] Srinivasan Parthasarathy, and Wei Li Mohammed Javeed Zaki, "A Localized Algorithm for Parallel Association Mining," in *In 9th ACM Symp. Parallel Algorithms & Architectures.* , 1997.